REVISION HISTORY			VARIATIONS FOR THIS ASSY.
CHK	ECO NO	REV	

FIRST USED ON: M9312	
MADE BY: B CRAMM	DATE: 1 AUG 78
CHECKED: N POLLITT	DATE: 16 AUG 78
DSN.ENG.: B GIST	DATE: 1 AUG 78
PROD.: D PETERSON	DATE: 3 AUG 78
RESP.ENG.: E CROCKER	DATE: 1 AUG 78

DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASSACHUSETTS

TITLE

ROM LISTING DIAGNOSTIC

SIZE	CODE	DOCUMENT NUMBER	REV
K	SP	M9312-0-5	B

ASSY. #:	EDIT NO
	1

IDENTIFICATION

PRODUCT NAME ROM LISTING DIAGNOSTIC (K-SP-M9312-0-5)

DATE 1 AUG 1978

SECTION 1

ROM 23-248F1

TABLE OF CONTENTS

1.0 OPERATIONAL NOTES

1.0 OPERATIONAL NOTES

1.1. OVERVIEW

THE M9312 IS DESIGNED TO PROVIDE BOOTSTRAPPING CAPABILITIES FOR ALL PDP-11 SYSTEMS WITH OR WITHOUT THE CONSOLE SWITCH REGISTER. IN ADDITION TO PROVIDING BOOTSTRAPPING FUNCTIONS FOR ALL MAJOR PDP-11 DEVICES, THE M9312 INCLUDES ROUTINES FOR CONSOLE EMULATION AND ALSO PROVIDES SOME BASIC CPU AND MEMORY GO-NOGO DIAGNOSTIC TESTS.

THIS BOOTSTRAP HAS BEEN DESIGNED FOR MAXIMUM FLEXIBILITY OF OPERATION. ITS FUNCTIONS MAY BE INITIATED AUTOMATICALLY AT A POWER UP, OR BY DEPRESSING THE CONSOLE BOOT SWITCH, OR BY A LOAD ADDRESS AND START SEQUENCE, OR EVEN BY USING THE CONSOLE TTY WHILE RUNNING THE CONSOLE EMULATOR.

1.2. INTERNAL SWITCH SETTING

A SET OF EIGHT MICRO SWITCHES ARE LOCATED ON THE M9312 MODULE. THESE ARE USED BY THE ROUTINES TO DETERMINE WHAT ACTION IS TO BE TAKEN. THEY GIVE THE USER AUTOMATIC ACCESS TO ANY FUNCTION.

A. POWER UP AND CONSOLE BOOT SWITCHES

THE PRIMARY ACTIVATING PROCESSES FOR THE M9312 PROGRAMS ARE EITHER A POWER UP SEQUENCE OR THE ENABLING OF THE CONSOLE BOOT SWITCH.

TO ACTIVATE THE M9312 ON A POWER UP, SWITCH 2 IN THE M9312 MICRO SWITCH REGISTER MUST BE IN THE ON POSITION. IF THIS SWITCH IS OFF THEN A NORMAL TRAP TO LOCATION 24 TO EXECUTE THE USER POWER UP ROUTINE WILL OCCUR. WHEN THIS SWITCH IS ON THE OTHER SWITCHES, 3 THROUGH 10, WILL DETERMINE WHAT ACTION THE M9312 WILL TAKE WHEN THE POWER UP OCCURS (SEE MICRO SWITCH SETTING BELOW).

IF THE SYSTEM INCLUDES A CONSOLE BOOT SWITCH THEN ANY TIME THAT SWITCH IS PRESSED THE M9312 WILL BE ACTIVATED (SOME PROCESSORS MAY HAVE TO BE HALTED FOR THIS SWITCH TO HAVE ANY EFFECT). THE PROCESS USED TO ENTER THE ROM IS A "FAKE" POWER DOWN FOLLOWED BY A POWER UP CAUSED BY PRESSING THE BOOT SWITCH (NOTE THAT THE POSITION OF MICRO SWITCH 2 DESCRIBED ABOVE IS IRRELIVANT TO THE OPERATION OF THIS BOOT SWITCH). THIS RESULTS IN A NORMAL POWER UP SEQUENCE IN THE CPU. PRIOR TO THE POWER UP SEQUENCE, THE M9312 ASSERTS 773000 ON THE UNIBUS ADDRESS LINES. THIS CAUSES THE NEW PC TO BE TAKEN FROM ROM LOCATION 773024 INSTEAD OF FROM LOCATION 000024. THE NEW PC WILL BE THE LOGICAL 'OR' OF THE CONTENTS OF ROM LOCATION 773024 AND THE EIGHT MICRO SWITCHES ON THE M9312 MODULE (A SWITCH IN THE ON POSITION IS SEEN AS A ONE; LIKEWISE A SWITCH IN THE OFF POSITION IS A ZERO).

IN THIS WAY ALL THE M9312 OPTIONS ARE ACCESSABLE BY MERELY GIVING EACH OPTION A DIFFERENT STARTING ADDRESS. NOTE HERE THAT MICRO SWITCH NUMBER 10 IS OR'ED WITH BIT 1 OF THE DATA IN ROM LOCATION 773024, MICRO SWITCH NUMBER NINE IS OR'ED WITH BIT 2 ETC., AND THAT IT IS UNNECESSARY TO PROVIDE A SWITCH WHICH IS OR'ED WITH DATA BIT 0 AS THIS COULD RESULT IN AN ODD ADDRESS WHEN GOING THROUGH THE TRAP TO LOCATION 773024 SEQUENCE.

B. THE MICRO SWITCH SETTINGS.

THE SETTING OF THE MICRO SWITCHS DEPENDS ON THE VARIOUS ROMS USED AND THE POSITION THEY ARE USED IN ON THE M9312.

1.3. CONSOLE EMULATOR

THESE ROUTINES ARE ESSENTIAL TO ANY PROCESSOR WITHOUT A CONSOLE. THEY PROVIDE THE USER THE CONSOLE FUNCTIONS OF LOAD ADDRESS, EXAMINE, DEPOSIT, AND START. ALSO THE ABILITY TO EXECUTE A BOOTSTRAP FUNCTION WITH ANY OF THE ABOVE DEVICES IS GIVEN.

THE FIRST THING THAT WILL EXECUTED (IF 020 IS THE CONTENTS OF THE MICRO SWITCHES) ARE THE PRIMARY CPU DIAGNOSTICS.

THEN THE DISPLAY ROUTINE IS ENTERED. THIS ROUTINE TYPE THE CONTENTS OF R0, R4, sp AND R5 (NOTE THE SEQUENCE!) ON THE TELETYPE AS FOUR 16 BIT OCTAL NUMBERS. PRESSING THE CONSOLE BOOT SWITCH CAUSES PDP-11 SYSTEMS (E.G. PDP-11/04'S) WITHOUT CONSOLE SWITCH REGISTERS TO COPY THE PC INTO R5 BEFORE THE POWER UP SEQUENCE STARTS.

A KEYBOARD DISPATCH ROUTINE IS THEN ENTERED TO INTERPRETE THE USERS COMMANDS. THIS ROUTINE TYPES A CARRIAGE RETURN AND A LINE FEED, THEN GIVES THE PROMPT '@'. ALL COMMANDS ARE TWO CHARACTERS. IF THE USER TYPES AN ILLEGAL COMMAND IT WILL BE IGNORED AND THE KEYBOARD DISPATCH IS RESTARTED. LEGAL COMMANDS ARE:

L<SPC>NUMBER
: LOAD ADDRESS::LOAD THE INTERNAL ADDRESS POINTER WITH 'NUMBER', A 16 BIT OCTAL NUMBER.

E<SPC>
: EXAMINE::EXAMINE AND DISPLAY ON THE TELETYPE THE ADDRESS AND THE CONTENTS OF THAT ADDRESS IN THE INTERNAL ADDRESS POINTER (SEE LOAD ADDRESS). NOTE THAT IF THE PREVIOUS COMMAND WAS ALSO AN EXAMINE COMMAND THEN THE INTERNAL ADDRESS POINTER IS INCREMENTED BY 2.

D<SPC>
: DEPOSIT::DEPOSIT THE VALUE NUMBER INTO THE LOCATION POINTED TO BY THE INTERNAL ADDRESS POINTER (SEE LOAD ADDRESS). IF THE PREVIOUS COMMAND WAS ALSO DEPOSIT THEN THE POINTER IS INCREMENTED BY 2.

S<CR>
: START::CAUSES A RESET INSTRUCTION TO BE EXECUTED AND A JMP TO THE LOCATION SPECIFIED IN THE INTERNAL ADDRESS POINTER.

DP#

BOOT RP02 OR RP03::LOAD AND EXECUTE THE MONITOR FROM THE DEVICE USING THE DRIVE NUMBER OPTIONALLY SPECIFIED BY # (DEFAULT 0) WITH OR WITHOUT FIRST RUNNING SECONDARY CPU AND MEMORY DIAGNOSTICS.

DB#

BOOT RP04/5/6 OR RM02/3::LOAD AND EXECUTE THE MONITOR FROM THE DEVICE USING THE DRIVE NUMBER OPTIONALLY SPECIFIED BY # (DEFAULT 0) WITH OR WITHOUT FIRST RUNNING SECONDARY CPU AND MEMORY DIAGNOSTICS.

DS#

BOOT RS03 OR RS04::LOAD AND EXECUTE THE MONITOR FROM THE DEVICE USING THE DRIVE NUMBER OPTIONALLY SPECIFIED BY # (DEFAULT 0) WITH OR WITHOUT FIRST RUNNING SECONDARY CPU AND MEMORY DIAGNOSTICS.

DK#

BOOT RK05 RK03/RK05J::LOAD AND EXECUTE THE MONITOR FROM THE DEVICE USING THE DRIVE NUMBER OPTIONALLY SPECIFIED BY # (DEFAULT 0) WITH OR WITHOUT FIRST RUNNING SECONDARY CPU AND MEMORY DIAGNOSTICS.

DM#

BOOT RK06/RK07:LOAD AND EXECUTE THE MONITOR FROM THE DEVICE USING THE DRIVE NUMBER OPTIONALLY SPECIFIED BY # (DEFAULT 0) WITH OR WITHOUT FIRST RUNNING SECONDARY CPU AND MEMORY DIAGNOSTICS.

DX#

BOOT RX01::LOAD AND EXECUTE THE MONITOR FROM THE DEVICE USING THE DRIVE NUMBER OPTIONALLY SPECIFIED BY # WHICH MUST BE 0 OR 1 (DEFAULT 0) WITH OR WITHOUT FIRST RUNNING SECONDARY CPU AND MEMORY DIAGNOSTICS.

```
DT#
        BOOT   TU56/TU55
        LOAD AND EXECUTE  THE
        MONITOR FROM THE  DEVICE  USING  THE
        DRIVE NUMBER OPTIONALLY SPECIFIED BY
        # (DEFAULT 0) WITH OR WITHOUT  FIRST
        RUNNING  SECONDARY  CPU  AND  MEMORY
        DIAGNOSTICS.

DY#
        BOOT RX02::LOAD AND START THE
        MONITOR FROM THE  DEVICE  USING  THE
        DRIVE NUMBER OPTIONALLY SPECIFIED BY
        # (DEFAULT 0) WITH OR WITHOUT FIRST
        RUNNING SECOUNDARY CPU AND MEMORY
        DIAGNOSTICS.

MT#
        BOOT   TU10 TE10,TS03
        LOAD AND EXECUTE  THE
        MONITOR FROM THE  DEVICE  USING  THE
        DRIVE NUMBER OPTIONALLY SPECIFIED BY
        # (DEFAULT 0) WITH OR WITHOUT  FIRST
        RUNNING  SECONDARY  CPU  AND  MEMORY
        DIAGNOSTICS.

MM#
        BOOT   TU16,TE16,TM02/3::
        LOAD AND EXECUTE  THE
        MONITOR FROM THE  DEVICE  USING  THE
        DRIVE NUMBER OPTIONALLY SPECIFIED BY
        # (DEFAULT 0) WITH OR WITHOUT  FIRST
         SECONDARY  CPU  AND  MEMORY
        DIAGNOSTICS.

TT
        BOOT  DL11::READ  THE ABSOLUTE PAPER
        TAPE LOADER FROM THE TELETYPE  PAPER
        TAPE  READER  WITH  OR WITHOUT FIRST
        RUNNING THE SECONDARY CPU AND MEMORY
        DIAGNOSTICS.

PR
        BOOT  PC11::READ  THE ABSOLUTE PAPER
        TAPE  LOADER  FROM  THE  HIGH  SPEED
        PAPER  TAPE  READER  WITH OR WITHOUT
        FIRST EXECUTING  SECONDARY  CPU  AND
        MEMORY DIAGNOSTICS.
```

```
DL#
         BOOT RL11/RL01
         LOAD AND EXECUTE  THE
         MONITOR FROM THE  DEVICE  USING  THE
         DRIVE NUMBER OPTIONALLY SPECIFIED BY
         # (DEFAULT 0) WITH OR WITHOUT  FIRST
         RUNNING  SECONDARY  CPU  AND  MEMORY
         DIAGNOSTICS.

CT#
         BOOT TU60:: LOAD AND EXECUTE THE
         MONITOR FROM THE DEVICE USEING
         THE DRIVE NUMBER OPTIONALLY
         SPECIFIED BY # (DEFAULT 0) WITH OR
         WITHOUT CPU AND MEMORY DIAGNOSTICS.



NOTE::  MORE DEVICES MAY BE ADDED AFTER THIS
     DOCUMENT IS RELEASED.  THE INSTALATION
     DOCUMENT(K-SP-M9312-0-4)WILL BE UPDATED
     AFTER EACH NEW ROM FOR A DEVICE WRITTEN
     YOU MAY REFERENCE THAT DOCUMENT.

     ALSO THE DIAGNOSTIC FOR THE M9312 WILL
     TYPE OUT ALL NUMONICS WITHIN THE M9312.
```

1.4. BOOTSTRAPPING

THESE ROUTINES TO BOOTSTRAP A DEVICE TYPICALLY READ IN THE FIRST SECTOR, BLOCK OR 512 (DEC) WORDS, OFF THE DEVICE INTO LOCATION 0 THROUGH 512 (DEC) OF MEMORY. THE EXCEPTIONS TO THIS RULE ARE THE PAPER TAPE BOOT, THE FLEXIBLE DISK BOOT AND THE MAGNETIC TAPE BOOTS. THE PAPER TAPE BOOT IS UNIQUE IN THAT IT CAN DO NO ERROR CHECKING AND THAT THE SECONDARY BOOTSTRAP (THE ABSOLUTE LOADER, FOR EXAMPLE) IS READ INTO THE UPPER PART OF MEMORY. THE ACTUAL LOCATION LOADED BY THE PAPER TAPE BOOT ARE PARTIALLY DETERMINED BY THE SECONDARY BOOTSTRAP ITSELF AND BY THE 'SIZE' ROUTINE WHICH DETERMINES THE HIGHEST AVAILABLE MEMORY ADDRESS WITHIN THE FIRST 28K. THE FLEXIBLE DISK (OR FLOPPY) READS SECTOR 1 ON TRACK 1 INTO LOCATIONS STARTING AT ZERO. THE MAGNETIC TAPE BOOTS READ THE SECOND BLOCK INTO LOCATIONS STARTING AT 0. IF NO ERRORS ARE DETECTED IN THE DEVICE, THE BOOTSTRAPS NORMALLY TRANSFER CONTROL TO LOCATION 0 IN ORDER TO EXECUTE THE SECONDARY BOOTSTRAP JUST LOADED. THE ONLY EXCEPTION TO THIS STARTING ADDRESS IS WITH THE PAPER TAPE BOOTS. THEY TRANSFER CONTROL TO LOCATION XXX374, WHERE XXX WAS DETERMINED INITIALLY BY THE SIZE ROUTINE TO BE AT THE TOP OF MEMORY; THIS IS WHERE THE ABSOLUTE LOADER WAS JUST LOADED.

IF A DEVICE ERROR IS DETECTED A RESET WILL BE EXECUTED AND THE BOOTSTRAP WILL TRY AGAIN. THE BOOTSTRAP WILL BE RETRIED INDEFINITELY UNTIL IT SUCCEEDS WITHOUT ERROR UNLESS THE USERS (OPERATOR) INTERVENES. THE ADVANTAGE OF RETRYING THE BOOT IS THAT IF A PARTICULAR DEVICE BEING BOOTED IS NOT ONLINE OR LOADED, SAY BECAUSE OF A POWER FAILURE RESTART, THE BOOT WILL GIVE THE DEVICE A CHANCE TO POWER UP (FOR DISKS THIS IS ESSENTIAL). A MAGNETIC TAPE TRANSPORT WILL NOT AUTOMATICALLY RELOAD ITSELF AFTER A POWER FAILURE AND RESTART. THIS SITUATION REQUIRES USER INTERVENTION. THE USER MUST RELOAD THE MAGTAPE AND BRING IT BACK ONLINE AT WHICH TIME THE MAGTAPE BOOTSTRAP, WHICH WAS CONTINUALLY ATTEMPTING TO BOOT THE TAPE, WILL SUCCEED.

NOTE:: AN EXCEPTION TO THIS RULE ARE DEVICE BOOTS FOR THE
RK06/RK07 AND RX02.
IN THE CASE OF THE RK06,IF A DRIVE TYPE ERROR IS
ENCOUNTERED,THE DRIVE IS SET TO AN RK07.
IN THE CASE OF AN RX02,FIRST DOUBLE DENSITY IS TRYED,
SOULD WE GET A DENSITY ERROR ON READ TRY,WE CHANGE
THE DENSITY TO SINGLE DENSITY. IN EITHER CASE,WE
READ TWO (2) SECTORS OF DATA.
SOME BOOTS ALLOW POWER-UP OR BOOT SWITCH BOOTING FOR
DRIVES OTHER THAN DRIVE 0. REFER TO INDIVIDUAL BOOTS FOR THIS
INFORMATION.

1.5. DIAGNOSTIC TESTS

THERE ARE THREE DIFFERENT TYPES OF TESTS INCLUDED IN THE M9312 :

1 PRIMARY CPU TESTS
2 SECONDARY CPU TESTS
3 MEMORY TEST

THE PRIMARY CPU TESTS ARE TESTS OF ALL UNARY AND DOUBLE OPERAND INSTRUCTIONS WITH ALL SOURCE MODES. THESE TESTS DO NOT MODIFY MEMORY. IF A FAILURE IS DETECTED A BRANCH DOT WILL BE EXECUTED.

THE SECONDARY CPU TESTS MODIFY MEMORY AND INVOLVE THE USE OF THE STACK POINTER. THESE TESTS INCLUDE TESTING OF THE JMP AND JSR INSTRUCTIONS AS WELL AS TESTS OF ALL DESTINATION MODES. IF A FAILURE IS DETECTED THESE TESTS, UNLIKE THE PRIMARY TESTS, WILL EXECUTE A HALT. THE USER MAY THEN CONSULT THIS LISTING TO DETERMINE THE FAULT CLASS FOR THE PARTICULAR LOCATION THE TEST HALTED.

FINALLY THE MEMORY TEST PERFORMS BOTH A DUAL ADDRESSING AND DATA CHECK OF ALL THE AVAILABLE MEMORY ON THE SYSTEM LESS THAN 28K. THIS TEST WILL LEAVE ALL OF MEMORY CLEAR. LIKE THE SECONDARY TESTS THE MEMORY TEST WILL HALT WHEN AN ERROR IS DETECTED. AT THE TIME THE MEMORY ERROR HALT IS EXECUTED R0 WILL CONTAIN THE ADDRESS AT WHICH THE FAILURE WAS DETECTED PLUS TWO; R4 WILL CONTAIN THE FAILING DATA PATTERN AND R6 WILL CONTAIN THE EXPECTED DATA PATTERN. THUS AFTER A MEMORY FAILURE HAS OCCURRED THE USER CAN ENTER THE CONSOLE EMULATOR AND HAVE THIS INFORMATION PRINTED OUT TO HIM IMMEDIATELY BY THE DISPLY ROUTINE (SEE ABOVE SECTION ON CONSOLE EMULATOR).

NOTE: HERE THAT DIAGNOSTICS ARE RUN OR NOT RUN (OPTIONALLY) DEPENDING ON WHETHER BIT 1 IS CLEAR OR SET (RESPECTIVELY) IN EITHER:

1 THE INTERNAL MICRO SWITCHES DESCRIBED ABOVE (IF THE POWER UP OR CONSOLE BOOT SWITCH METHOD IS USED); OR
2 THE CONSOLE SWITCH REGISTER (IF THE LOAD ADDRESS AND START WITH OPTION CODE IN SWITCHES METHOD DESCRIBED BELOW IS USED).

1.6. RESTARTING AT THE USER POWER FAIL ROUTINE

IF THE USER WISHES TO RESTART HIS OWN SOFTWARE ON A POWER UP HE MAY DO SO BY MERELY DISABLING THE POWER FAIL RESTART SWITCH IN THE MICRO SWITCHES (TURN SWITCH 2 OFF).

BUT THE USER CAN USE THE M9312 TO RUN DIAGNOSTICS (JUST THE PRIMARY CPU TESTS DESCRIBED ABOVE) BEFORE RUNNING HIS POWER UP ROUTINE. THIS WILL IN NO WAY DISTURB THE CONTENTS OF MEMORY AND WILL IN FACT VARIFY THE MACHINE'S BASIC INTEGRITY AFTER THE POWER DOWN AND UP SEQUENCE.

TO USE THIS OPTION PUT THE CODE 644 INTO THE MICRO SWITCHES AS DESCRIBED ABOVE. ALSO SWITCH 2 MUST BE OFF. THIS WILL RESULT IN THE RUNNING OF THE PRIMARY CPU DIAGNOSTICS AND THEN A SIMULATED TRAP THROUGH 24 WHICH WILL START THE USER'S SPECIFIED POWER UP ROUTINE.

IF THE CODE 646 IS PLACED IN THE INTERNAL SWITCHES THEN THE SIMULATED TRAP THROUGH 24 IS EXECUTED WITHOUT RUNNING ANY DIAGNOSTICS.

1.7. LOAD ADRESS AND START PROCEDURE

ASCII CONSOLE AND DIAGNOSTIC ROM FOR USE WITH 11/04/05/10/20/34/35/40/45/50/55

FUNCTION TO BE BOOTED	ROM LOC.	DIAG-NOSTIC	MANUAL STARTING ADDR.
CONSOLE EMULATOR	E20	NO	165144
CONSOLE EMULATOR	E20	YES	165020

THE CONSOLE EMULATOR CAN PERFORM THE FOLLOWING FUNCTIONS:

1. LOAD ADDRESS
2. EXAMINE
3. DEPOSIT
4. START
5. BOOT A DEVICE

THE EXAMPLE BELOW CONTAINS THE START ADDRESS FOR EACH OF THE 4 ROM LOCATIONS REFERENCE THE INSTALLATION AND SET UP PROCEDURE (K-SP-M9312-0-4)FOR A COMPLETE LISTING

DEVICE TO BE BOOTED	ROM LOC.	DIAG-NOSTIC	UNIT	START ADDR.
RL01	ROM 1	NO	0	173004
RL01	ROM 1	YES	0	173006
RL01	ROM 2	NO	0	173204
RL01	ROM 2	YES	0	173206
RL01	ROM 3	NO	0	173404
RL01	ROM 3	YES	0	173406
RL01	ROM 4	NO	0	173604
RL01	ROM 4	YES	0	173606

2.0 PRIMARY CPU DIAGNOSTICS

2.1 COMMON DATA TABLE

2.2 TEST 1 SINGLE OPERAND INSTRUCTIONS

2.3 TEST 2 DOUBLEOPERAND INSTRUCTIONS, ALL SOURCE MODES, DEST MODE 0

2.4 TEST 3 JMP TEST MODES 1,2AND 3

2.5 TEST 4 SINGLE OPERAND,NON-MODIFING INSTRUCTION, BYTE REFERENCING TEST

2.1 THIS IS A TABLE OF DATA USED IN BOTH THE PRIMARY AND THE SECONDARY CPU DIAGNOSTICS.

```
165000  165000                  POOL:   .WORD   POOL
165002  165000                          .WORD   POOL
165004  100000                  T4DATA: .WORD   100000
165006  177777                  T6DATA: .WORD   177777
165010  165006                          .WORD   T6DATA
165012  165006                          .WORD   T6DATA
165014  000500                  T6DAT1: .WORD   500
165015  000501                          .WORD   501
```

2.2 ********** TEST 1 SINGLE OPERAND INSTRUCTIONS *********

```
165020  005003                  TEST1:  CLR     R3              ;       000000          0100
165022  005203                          INC     R3              ;       000001          0000
165024  005103                          COM     R3              ;       177776          1001
165026  006203                          ASR     R3              ;       177777          1010
165030  006303                          ASL     R3              ;       177776          1001
165032  006003                          ROR     R3              ;       177777          1010
165034  005703                          TST     R3              ;       177777          1000
165036  005403                          NEG     R3              ;       000001          0001
165040  005303                          DEC     R3              ;       000000          0101
165042  005603                          SBC     R3              ;       177777          1001
165044  006103                          ROL     R3              ;       177777          1001
165046  005503                          ADC     R3              ;       000000          0101
165050  000303                          SWAB    R3              ;       000000          0100
165052  001377                          BNE     .               ;ERROR IF NOT ZERO.
```

2.3 ********** TEST 2 DOUBLE OPERAND INSTRUCTIONS, ALL SOURCE MODES, DEST MODE 0. **********

```
165054  012702  165000          TEST2:  MOV     #POOL,R2        ;SET UP ADDRESS.
165060  011203                          MOV     (R2),R3         ;MOVE FROM TABLE DEFFERRED.
165062  022203                          CMP     (R2)+,R3        ;CHECK FOR CORRECT DATA.
165064  001377                          BNE     .               ;LOOP HERE IF NOT EQUAL.
165066  063203                          ADD     @(R2)+,R3       ;ADD TO REGISTER.
165070  165203                          SUB     @-(R2),R3       ;SUBTRACT SAME DATA FROM REGISTER.
165072  044203                          BIC     -(R2),R3        ;SHOULD CLEAR REGISTER.
165074  056203  000012                  BIS     12(R2),R3       ;SHOULD SET ALL REGISTER'S BITS.
165100  037203  000012                  BIT     @12(R2),R3      ;SHOULD TEST NOT ZERO (Z=0).
165104  001777                          BEQ     .               ;LOOP HERE ON ERROR.
```

2.4 ********** TEST 3 JMP TEST MODES 1, 2 AND 3. **********

```
165106  010703                  TEST3:  MOV     PC,R3           ;SET UP ADDRESS FOR MODE 1 JMP.
165110  000123                  JMP1:   JMP     (R3)+           ;EXECUTE JMP MODE 2.
165112  012703  165122                  MOV     #JMP2A,R3       ;SET UP ADDRESS OF ADDRESS FOR MODE 3 JMP.
165116  000133                          JMP     @(R3)+          ;EXECUTE JMP MODE 3.
165120  000113                  JMP2:   JMP     (R3)            ;EXECUTE MODE 1 JMP TO NEXT TEST.
165122  165120                  JMP2A:  .WORD   JMP2
```

2.5 ********** TEST 4 SINGLE OPERAND, NON-MODIFYING INSTRUCTION, BYTE REFERENCING TEST. **********

```
165124  105767  177654        TEST4:  TSTB    T4DATA          ;TEST EVEN BYTE.
165130  001377                        BNE     .               ;LOOP ON ERROR, NOT ZERO.
165132  022222                        CMP     (R2)+,(R2)+     ;GET ADDRESS OF T4DATA IN R2.
165134  105722                        TSTB    (R2)+           ;EXAMINE DATA USING MODE 2.
165136  001377                        BNE     .               ;LOOP IF NOT ZERO.
165140  105712                        TSTB    (R2)            ;TST ODD BYTE MODE 1.
165142  100377                        BPL     .               ;LOOP IF POSITIVE.
```

3.0 CONSOLE FUNCTIONS

3.1 DISPLAY

3.2 START

3.3 DEPOSIT

3.4 LOAD ADDRESS

3.5 EXAINE

3.0 CONSOLE FUNCTIONS

3.1 ********** DISPLAY *********

```
165144  010701                  DSPLY:  MOV     PC,R1
165146  000554                          BR      PUTCR           ;PRINT CR, LF AND FILLER CHARACTERS.
165150  010701                          MOV     PC,R1
165152  000526                          BR      PUTNUM          ;PRINT R0.
165154  010400                          MOV     R4,R0
165156  000524                          BR      PUTNUM          ;PRINT R4.
165160  010600                  OUT:    MOV     SP,R0
165162  010701                          MOV     PC,R1
165164  000521                          BR      PUTNUM          ;PRINT R6.
165166  010500                          MOV     R5,R0
165170  000517                          BR      PUTNUM          ;PRINT R5.
165172  010605                          MOV     SP,R5
165174  010701                  CONSEM: MOV     PC,R1
165176  000540                          BR      PUTCR           ;PRINT CR, LF AND FILLER CHARACTERS.
165200  112702  000100                  MOVB    #100,R2         ;44 (OCTAL) IS THE ASCII FOR '$', THE
165204  010703                          MOV     PC,R3           ;COMMAND PROMPT CHARACTER.
165206  000554                          BR      PUTCHR
165210  010706                  READ:   MOV PC,SP
165212  000544                          BR      GETCHR          ;GET THE FIRST CHARACTER OF A COMMAND.
165214  000302                          SWAB    R2
165216  000542                          BR      GETCHR          ;GET THE SECOND CHARACTER OF THE COMMAND.
165220  020227                          CMP     R2,(PC)+        ;LOAD ADDR. INSTRUCTION?
165222  046040                          .ASCII  " L"
165224  001450                          BEQ     LA
165226  020402                          CMP     R4,R2           ;EQUALS LAST COMMAND?
165230  001001                          BNE     1$              ;NO SKIP.
165232  005725                          TST     (5)+            ;YES-UPDATE ADDR. POINTER.
165234  010204                  1$:     MOV     R2,R4           ;RECORD THIS COMMAND.
165236  020227                          CMP     R2,(PC)+        ;IS CMMD A EXAM?
165240  042440                          .ASCII  ' E'
165242  001446                          BEQ     EX              ;IF SO DO EXAM FUNCTION.
165244  020227                          CMP     R2,(PC)+        ;IS CMMD A DEPOSITE?
165246  042040                          .ASCII  ' D'
165250  001432                          BEQ     DE              ;IF SO DO DEPOSITE FUNCTION.
165252  020227                          CMP     R2,(PC)+        ;IS IT A START COMMAND?
165254  051415                          .ASCII  <CR>'S'
```

3.0 CONSOLE FUNCTIONS

3.2 *********** START **********

```
165256  001002                       BNE     2$
165260  000005                       RESET
165262  000115                       JMP     (R5)
165264  012704  173000       2$:     MOV      173000,R4
165270  031427  000200       3$:     BIT     (R4),#200          ;NO ROM?
165274  001323                       BNE     DSPLY              ;YEA,GO BACK,ABORT SEARCH.
165276  022402                       CMP     (R4)+,R2           ;MATCH ON BOOT HEADER VERSES CMMD?
165300  001405                       BEQ     4$                 ;YES!--GO AHEAD.
165302  061404                       ADD     (R4),R4            ;NO-ADD INDEX TO GET TO NEW BOOT HEADER.
165304  020427  174000               CMP     R4, 174000         ;HAVE WE EXCEEDED ROM RANGE?
165310  001731                       BEQ     CONSEM
165312  000766                       BR      3$                 ;LOOP UNTILL MATCH OR TRAPS OUT.
165314  010701               4$:     MOV     PC,R1
165316  000423                       BR      GETNUM
165320  000005                       RESET                      ;INIT SYSTEM
165322  113705  173024               MOVB    @#INITSW,R5        ;GET MICRO SWITCHES.
165326  106105                       ROLB    R5                 ;ASSUMES HERE THAT THAE STARTING
165330  106105                       ROLB    R5                 ;ADDR. OF DISPLY IS 165144
                                                                ;SO C BIT GETS SET.
165332  000164  000010       5$:     JMP     10(R4)             ;GO DO BOOT.
```

3.3 ********** DEPOSIT **********

```
165336  010701               DE:     MOV     PC,R1
165340  000412                       BR      GETNUM             ;GET INPUT DATA.
165342  010015                       MOV     R0,(R5)            ;PUT INPUT DATA IN MEMORY.
165344  000713                       BR      CONSEM             ;RETURN FOR NEXT COMMAND.
```

3.4 ********** LOAD ADDRESS **********

```
165346  010701               LA:     MOV     PC,R1
165350  000406                       BR      GETNUM             ;GET INPUT DATA.
165352  010005                       MOV     R0,R5              ;LEAVE IT AS THE CURRENT ADDRESS
                                                                ;POINTER IN R5.
165354  005004               KBDCLR: CLR     R4                 ;CLEAR THE PREVIOUS COMMAND FLAG.
165356  000706                       BR      CONSEM             ;RETURN FOR NEXT COMMAND.
```

3.5 ********** EXAMINE **********

```
165360  010506               EX:     MOV     R5,SP              ;USE THE DISPLAY ROUTINE TO PRINT
165362  011505                       MOV     (R5),R5            ;OUT THE ADDRESS, R5, AND THE DATA IN THAT
165364  000675                       BR      OUT                ;ADDRESS, (R5).
```

4.0 TTY HANDLERS

4.0 TTY HANDLERS

4.1 ********** OCTAL NUMBER INPUT ROUTINE **********

```
;THIS ROUTINE IS CALLED BY PLACING THE RETURN ADDRESS MINUS TWO
;IN R1 AND THEN BRANCHING TO GETNUM. THE ROUTINE WILL ATTEMPT TO
;INPUT A STRING OF CHARACTERS FROM THE TELETYPE, TERMINATED BY CR,
;AND ASSEMBLE THEM AS A 16 BIT OCTAL NUMBER IN R0. IF AN ILLEGAL
;CHARACTER IS TYPED IN THE STRING, NOT AN
;OCTAL DIGIT, THE CONSOLE EMULATOR WILL BE RESTARTED. THE PREVIOUS
;CONTENTS OF R2 AND R3 ARE LOST. THE ROUTINE WILL RETURN WITH THE
;OCTAL NUMBER IN R0, TO THE ADDRESS SPECIFIED IN R1 PLUS TWO.
;BUT NOTE THAT R1 WILL HAVE BEEN INCREMENTED BY FOUR UPON RETURN.
```

```
165366  005000                    GETNUM: CLR     R0              ;INITIALIZE R0.
165370  005002                    2$:     CLR     R2              ;INITIALIZE R2.
165372  010703                            MOV     PC,R3           ;GO GET A CHARACTER.
165374  000453                            BR      GETCHR
165376  120227  000015                    CMPB    R2,#CR          ;IS IT CR?
165402  001433                            BEQ     X1RTN           ;IF YES RETURN WITH THE NUMBER IN R0.
165404  162702  000070                    SUB     _ '8,R2         ;IS THE CHARACTER A LEGAL OCTAL DIGIT?
165410  062702  000010                    ADD     #'8-'0,R2
165414  103357                            BCC     KBDCLR          ;IF NOT GO RESTART THE CONSOLE EMULATOR.
165416  006300                            ASL     R0              ;OTHERWISE ASSEMBLE THE NUMBER.
165420  006300                            ASL     R0
165422  006300                            ASL     R0
165424  050200                            BIS     R2,R0
165426  000760                            BR      2$              ;LOOP BACK FOR NEXT CHARACTER.
```

4.0 TTY HANDLERS

4.2 ********* OCTAL NUMBER OUTPUT ROUTINE **********

```
                                        ;THIS ROUTINE, PUTNUM, IS CALLED BY PLACING THE RETURN ADDRESS
                                        ;MINUS TWO IN R1 AND THE 16 BIT OCTAL NUMBER TO BE TYPED
                                        ;IN R0. PUTNUM WILL TYPE 6 CHARCTERS COMPRISING THE OCTAL DIGITS
                                        ;OF THE NUMBER IN R0 FOLLOWED BY ONE SPACE CHARCTER. THE PREVIOUS
                                        ;CONTENTS OF R2 AND R3 ARE LOST. RETURN IS MADE TO THE ADDRESS
                                        ;IN R1 PLUS TWO, BUT R1 IS INCREMENTED BY FOUR.

165430  012702                          PUTNUM: MOV     (PC)+,R2        ;COMPUTE FIRST DIGIT.
165432  000030                                  .WORD   <'0>/2
165434  000261                                  SEC
165436  006100                          2$:     ROL     R0              ;SHIFT NUMBER INTO R2 TO COMPUTE NEXT CHARACTER.
165440  106102                                  ROLB    R2
165442  010703                                  MOV     PC,R3           ;PRINT CHARACTER.
165444  000435                                  BR      PUTCHR
165446  012702                                  MOV     (PC)+,R2        ;GET READY TO COMPUTE NEXT CHARACTER.
165450     206                                  .BYTE   200!<'0/10>
165451     040                                  .BYTE   <' >
165452  006300                          4$:     ASL     R0              ;SHIFT NUMBER AND SEE IF DONE.
165454  001403                                  BEQ     6$              ;BRANCH IF DONE.
165456  106102                                  ROLB    R2
165460  103774                                  BCS     4$
165462  000765                                  BR      2$
165464  000302                          6$:     SWAB    R2              ;WHEN DONE COME HERE TO PRINT SPACE.
165466  010703                                  MOV     PC,R3
165470  000423                                  BR      PUTCHR
165472  022121                          X1RTN:  CMP     (R1)+,(R1)+     ;AND RETURN.
165474  000161  177776                          JMP     -2(R1)
```

4.3 ********* CR, LF AND FILLER CHARACTER OUTPUT ROUTINE **********

```
                                        ;THIS ROUTINE IS CALLED TO TYPE A CARRIAGE RETURN, LINE FEED FOLLOWED
                                        ;BY 12 FILLER CHARACTERS. THE CALL IS MADE BY FIRST PLACING THE RETURN
                                        ;ADDRESS MINUS TWO IN R1 AND EXECUTING BR PUTCR.
                                        ;THE PREVIOUS CONTENTS OF BOTH REGISTERS R2 AND R3 ARE LOST.
                                        ;WHEN FINISHED THIS SUBROUTINE WILL RETURN TO THE ADDRESS SPECIFIED
                                        ;IN R1 PLUS TWO, BUT R1 WILL BE INCREMENTED BY FOUR.

165500  012702                          PUTCR:  MOV     (PC)+,R2
165502  014012                                  .WORD   14012
165504  010703                          1$:     MOV     PC,R3
165506  000414                                  BR      PUTCHR
165510  061702                                  ADD     (PC),R2
165512  003767                                  BLE     X1RTN
165514  105002                                  CLRB    R2
165516  152702  000015                          BISB    _CR,R2
165522  000770                                  BR      1$
```

4.0 TTY HANDLERS

4.4 ********* CHARACTER INPUT ROUTINE **********

```
                              ;THIS ROUTINE IS CALLED TO BOTH INPUT AND ECHO A CHARACTER FROM
                              ;THE TTY. THE CHARACTER IS SAVED IN THE LOW BYTE OF R2. THE
                              ;HIGH BYTE OF R2 IS NOT MODIFIED. A CALL TO THIS
                              ;SUBROUTINE IS MADE BY FIRST PLACING THE RETURN ADDRESS MINUS 2 IN R3
                              ;AND THEN EXECUTING A BRANCH TO GETCHR. WHEN FINISHED
                              ;A RETURN WILL BE MADE TO THE ADDRESS SPECIFIED IN R3 PLUS TWO, BUT
                              ;R3 WILL HAVE BEEN INCREMENTED BY FOUR.

165524  105737  177560        GETCHR: TSTB    @#TKS           ;WAIT FOR KEYBOARD INPUT, READY.
165530  100375                        BPL     GETCHR
165532  105002                        CLRB    R2              ;CLEAR THE LOW BYTE OF R2.
165534  153702  177562                BISB    @_ TKB,R2       ;PLACE THE CHARACTER IN THE LOW BYTE OF R2.
                                                              ;PROCEED ON TO PUTCHR IN ORDER TO
                                                              ;ECHO THE CHARACTER IN THE LOW BYTE
                                                              ;OF R2.
```

4.5 ********* CHARACTER OUTPUT ROUTINE **********

```
                              ;THIS SUBROUTINE IS CALLED TO PRINT A CHARACTER. THE CHARACTER
                              ;MUST BE PLACED IN THE LOW ORDER BYTE OF R2. TO CALL THIS ROUTINE
                              ;PLACE THE RETURN ADDRESS MINUS TWO IN R3 AND BRANCH TO
                              ;PUTCHR. THE CONTENTS OF R2 ARE NOT MODIFIED. RETURN IS MADE
                              ;TO THE ADDRESS SPECIFIED IN R3 PLUS TWO, BUT R3 IS LEFT INCREMENTED
                              ;BY FOUR.

165540  105737  177564        PUTCHR: TSTB    @#TPS           ;WAIT FOR PRINTER READY.
165544  100375                        BPL     PUTCHR
165546  110237  177566                MOVB    R2,@#TPB        ;WHEN READY PRINT THE CHARACTER.
165552  142702  100200                BICB    #100200,R2      ;CLEAR ANY PARITY BITS THAT MAY BE SET.
                              ;THIS ROUTINE IS USED TO MAKE RETURNS FROM SUBROUTINES WHICH USE
                              ;R3 AS AN ADDRESS LINK. THOSE SUBROUTINES ARE CALLED
                              ;BY PLACING THE RETURN ADDRESS MINUS 2 IN R3. WHEN THE SUBROUTINE
                              ;IS FINISHED PROCESSING IT WILL BRANCH TO X3RTN. X3RTN
                              ;WILL INCREMENT R3 BY FOUR BUT RETURN TO THE ADDRESS SPECIFIED IN R3
                              ;PLUS TWO.
165556  022323                X3RTN:  CMP     (R3)+,(R3)+     ;INCREMENT R3 BY 4.
165560  000163  177776                JMP     -2(R3)          ;RETURN TO THE ADDRESS WHICH WAS
                                                              ;ORIGINALLY IN R3 PLUS TWO.
```

4.6 ********* REGISTER DISPLAY ROUTINE **********

```
                              ;DSPLY IS THE FIRST PART OF THE CONSOLE EMULATOR THAT IS EXECUTED.
                              ;IT WILL PRINT THE CONTENTS OF R0, R4, R6 AND R5 (IN THAT ORDER) AS
                              ;FOUR SIXTEEN BIT OCTAL NUMBERS ON THE TTY. NOTE THAT ON SOME ELEVEN
                              ;SYSTEMS (E.G. THE 11/04) WHEN THE BOOT SWITCH IS DEPRESSED AND
                              ;THE PROCESS OF BOOTING INITIATED THE PC WILL BE PLACED INTO R5.
```

5.0 SECONDARY CPU DIAGNOSTICS

5.1 DOUBLE OPERAND, MODIFYING INSTRUCTIONS, BYTE REFERENCE TEST

5.2 JSR TEST WITH MODES 1 AND 6

5.0 SECONDARY CPU DIAGNOSTICS

```
;THESE ARE THE MEMORY MODIFYING TESTS. THEY ARE RUN AS SUBROUTINE
;WHICH IS CALLED BY PLACING THE RETURN ADDRESS IN THE CALLING
;ROUTINE MINUS TWO IN R1. THE PREVIOUS CONTENTS OF R2,
;R3, R5 AND R6 ARE LOST. IF ANY OF THESE TESTS DETECTS AN ERROR
;A HALT WILL BE EXECUTED. THE USER CAN THEN CONSULT THIS LISTING
;FOR FURTHER INFORMATION ABOUT THE FAULT. ISSUING A
;CONTINUE FUNCTION WILL NOT BE MEANINGFUL AFTER
;ANY OF THESE TESTS DETECTS AN ERROR AND HALTS. NOTE THAT
;WHEN THE MEMORY TEST DETECTS A FAILURE AND HALTS CERTAIN
;IMPORTANT PARAMETERS USED IN THAT TEST WILL BE SAVED IN THE
;REGISTERS. THESE REGISTERS ARE INCLUDED IN THOSE WHICH ARE
;DISPLAYED WHEN THE CONSOLE EMULATOR PORTION OF THIS ROM
;IS INITIATED. SO THAT IF A MEMORY ERROR IS DETECTED THE USER
;CAN START THE CONSOLE EMULATOR TO EASILY ACCESS THIS MEMORY
;ERROR DATA. CAUTION SHOULD BE OBSERVED WHEN SELECTING THESE
;TESTS TO BE RUN. THEY WILL DESTROY THE PREVIOUS CONTENTS
;OF ANY MAIN MEMORY AVAILABLE UP TO 28K WORDS.
```

5.1 ********* DOUBLE OPERAND, MODIFYING INSTRUCTIONS, BYTE REFERENCE TEST. **********

```
165564  012705  165006        TEST6:  MOV     #T6DATA,R5        ;SET UP TEST DATA ADDRESS.
165570  012702  000500                MOV     #500,R2           ;SET UP DEST ADDRESS.
165574  011503                        MOV     (R5),R3           ;SET UP CMP OPERAND FOR TEST.
165576  005012                        CLR     (R2)              ;CLR 500.
165600  112512                        MOVB    (R5)+,(R2)        ;500=000377.
165602  005202                        INC     R2                ;POINTER TO UPPER BYTE, 501.
165604  112512                        MOVB    (R5)+,(R2)        ;500=177777.
165606  005302                        DEC     R2                ;MOVE   POINTER BACK TO LOW BYTE.
165610  023512                        CMP     @(R5)+,(R2)       ;CHECK FOR ALL ONES.
165612  001015                        BNE     T6ERR
165614  005202                        INC     R2                ;MOVE POINTER UP TO ODD BYTE.
165616  143522                        BICB    @(R5)+,(R2)+      ;CLEAR HIGH BYTE OF LOCATION 500.
165620  024542                        CMP     -(R5),-(R2)       ;MOVE POINTERS BACK.
165622  143522                        BICB    @(R5)+,(R2)+      ;CLEAR LOW BYTE OF LOCATION 500.
165624  001010                        BNE     T6ERR             ;BRANCH IF NOT ZERO.
165626  010502                        MOV     R5,R2             ;GET DEFFERRED ADDRESS OF 500 INTO R2.
165630  016505  177772                MOV     -6(R5),R5         ;GET TEST DATA.
165634  110532                        MOVB    R5,@(R2)+         ;500 SHOULD BE 000377.
165636  150572  000000                BISB    R5,@(R2)          ;SET UPPER BITS.
165642  020352                        CMP     R3,@-(R2)         ;CHECK FOR ALL ONES, 177777.
165644  001407                        BEQ     TEST7             ;IF NO ERROR GO TO NEXT TEST.
165646  000000                T6ERR:  HALT
```

5.0 SECONDARY CPU DIAGNOSTICS

5.2 ********* JSR TEST WITH MODES 1 AND 6. **********

```
165650  005723                    TSTJSR: TST     (R3)+           ;R3 SHOULD POINT TO ZERO WORD AT T7ERR.
165652  001011                            BNE     T7ERR           ;IF NOT THEN HALT ON ERROR.
165654  021605                            CMP     (SP),R5         ;SP SHOULD POINT TO 177777.
165656  001007                            BNE     T7ERR           ;IF NOT THEN HALT ON ERROR.
165660  000203                            RTS     R3
165662  000000                            HALT
165664  011206                    TEST7:  MOV     (R2),SP         ;SET UP STACK TO START AT 502.
165666  012702  165650                    MOV     #TSTJSR,R2      ;SET UP ADDRESS FOR JSR.
165672  005726                            TST     (SP)+           ;MOVE SP UP ONE WORD.
165674  004312                            JSR     R3,(R2)         ;EXECUTE JSR TO TSTJSR.
165676  000000                    T7ERR:  HALT
165700  004362  000004                    JSR     R3,4(R2)        ;EXECUTE JSR TO TSTJSR+4.
```

6.0 MEMORY DIAGNOSTIC AND SIZE ROUTINES

6.0 MEMORY DIAGNOSTIC AND SIZE ROUTINES

```
                                    ;THIS ROUTINE PERFORMS BOTH A DUAL ADDRESSING TEST AND A RUDIMENTARY
                                    ;DATA INTEGRITY TEST OF ANY MEMORY AVAILABLE UP TO 28K WORDS.
                                    ;FIRST EVERY LOCATION IS WRITTEN WITH ITS OWN ADDRESS, STARTING
                                    ;FROM LOCATION 0 AND WORKING UP. THEN AFTER EVERY LOCATION HAS
                                    ;BEEN WRITTEN WITH ITS OWN ADDRESS, THE ROUTINE RETURNS TO LOCATION
                                    ;0 AND NEGATES ITS CONTENTS. THEN THE
                                    ;ADDRESS OF THE LOCATION WHICH WAS JUST NEGATED IS ADDED TO
                                    ;THAT LOCATION. THE RESULT SHOULD BE ZERO. THE PROCESS IS CONTINUED
                                    ;FROM 0 TO THE TOP OF MEMORY. IF NO ERRORS ARE DETECTED MEMORY
                                    ;WILL BE LEFT FILLED WITH ZEROES. IF AN ERROR IS DETECTED THEN;
                                    ;       1       THE ADDRESS BEING TESTED IS LEFT IN R4
                                    ;       2       THE EXPECTED DATA FROM THAT LOCATION IS IN R6
                                    ;       3       THE FAILING DATA IS IN R0
                                    ;       4       FINALLY THE HALT AT LOCATION 165776 IN THIS
                                    ;               ROM IS EXECUTED.

165704                              MEMTST:
165704  012705  160000              SIZE:   MOV     #160000,R5      ;SET UP R5.
165710  005037  000006                      CLR     @#6             ;SET UP TRAP VECTOR 4.
165714  012737  165722  000004              MOV     #1$,@#4
165722  012706  000502              1$:     MOV     #502,SP         ;SET UP THE STACK POINTER.
165726  005745                              TST     -(R5)           ;MAKE THE REFERENCE AND DECREMENT R5.
165730  005003                              CLR     R3              ;R3 IS A POINTER USED TO READ AND WRITE MEMORY.
165732  010313                      2$:     MOV     R3,(R3)         ;WRITE THE CONTENTS OF R3 INTO THE LOCATION
                                                                    ;ADDRESSED BY R3.
165734  005723                              TST     (R3)+           ;INCREMENT R3 BY 2.
165736  020305                              CMP     R3,R5           ;R5 CONTAINS THE LAST AVAILABLE MEMORY
                                                                    ;ADDRESS UNDER 28K WORDS. SEE IF
                                                                    ;THE END OF MEMORY HAS BEEN REACHED.
165740  101774                              BLOS    2$              ;LOOP UNTIL DONE.
165742  005003                              CLR     R3              ;RETURN R3 TO LOCATION 0.
165744  005413                      4$:     NEG     (R3)            ;NEGATE THE LOCATION REFERENCED BY R3.
165746  060313                              ADD     R3,(R3)         ;ADD THE CONTENTS OF R3.
165750  005723                              TST     (R3)+           ;INCRMENT THE POINTER, R3, BY 2; AND SEE IF THE
                                                                    ;RESULT OF THE ADDITION WAS 0.
165752  001004                              BNE     MEMERR          ;IF NOT 0, THEN AN ERROR HAS OCCURRED,
165754  020305                              CMP     R3,R5           ;OTHERWISE CONTINUE UP UNTIL THE TOP
                                                                    ;OF MEMORY IS REACHED.
165756  101772                              BLOS    4$
165760  000164  000002              ENDTST: JMP     2(R4)
                                                                    ;RESULT IN A TRAP THEN RETURN
                                                                    ;WITH THE HIGHEST MEMORY ADDRESS
                                                                    ;AVAILABLE IN R5.
                                    ;IF A MEMORY ERROR IS DETECTED COME HERE TO SAVE IMPORTANT
                                    ;PARAMETERS IN THE REGISTERS AND HALT.

165764  014304                      MEMERR: MOV     -(R3),R4
165766  010300                              MOV     R3,R0
165770  005006                              CLR     SP
165772  000000                              HALT
165774  040460                              .ASCII  "0A"            ;IDENTIFIES ROM AS "A0" OR A VERSION 0 ECB 12-12-77
  1012  165776  123162                              .WORD   123162          ;CRC WORD FOR LAST 255. WORDS.
  1013          000001                              .END
```

```
                              248F1.P11 SYMBOL TABLE

BIT8   = 000400        BIT9   = 001000        CONSEM  165174        CR     = 000015
CRCWD  = 000000        DE       165336        DIAG   = 165564       DL11CR= 177560
DSPLY    165144        ENDTST   165760        EX       165360       GETCHR  165524
GETNUM   165366        HBOOT    165000        HSRCR  = 177550       INITSW= 173024
JMP1     165110        JMP2     165120        JMP2A    165122       KBDCLR  165354
LA       165346        LF     = 000012        MEMERR   165764       MEMTST  165704
MRESER= 173000         OUT      165160        PC     =%000007       PC11CR= 177550
PDIAG    165564        POOL     165000        PSW    = 177776       PUTCHR  165540
PUTCR    165500        PUTNUM   165430        READ     165210       RESERV= 000340
RK05CR= 177404         RK06CR= 177440         RL01CR= 174400        RP03CR= 176714
RP04CR= 176700         RS03CR= 172040         RS04CR= 172040        RX01CR= 177170
RX02CR= 177170         R0     =%000000        R1     =%000001       R2     =%000002
R3     =%000003        R4     =%000004        R5     =%000005       R6     =%000006
R7     =%000007        SIZE     165704        SP     =%000006       SWR    = 177570
TEST1    165020        TEST2    165054        TEST3    165106       TEST4    165124
TEST6    165564        TEST7    165664        TKB    = 177562       TKS    = 177560
TPB    = 177566        TPS    = 177564        TSTJSR   165650       TTCR   = 177560
TU10CR= 172522         TU16CR= 172440         TU56CR= 177342        T4DATA   165004
T6DATA   165006        T6DAT1   165014        T6ERR    165646       T7ERR    165676
X1RTN    165472        X3RTN    165556        .      = 166000
```

SECTION 2

ROM 23-616F1

TABLE OF CONTENTS

OPERATIONAL NOTES

OPERATIONAL NOTES

1. OVERVIEW

THE M9312 IS DESIGNED TO PROVIDE BOOTSTRAPPING CAPABILITIES FOR ALL PDP-11 SYSTEMS WITH OR WITHOUT THE CONSOLE SWITCH REGISTER. IN ADDITION TO PROVIDING BOOTSTRAPPING FUNCTIONS FOR ALL MAJOR PDP-11 DEVICES, THE M9312 INCLUDES ROUTINES FOR SOME BASIC CPU AND MEMORY GO-NOGO DIAGNOSTIC TESTS.

THIS BOOTSTRAP HAS BEEN DESIGNED FOR MAXIMUM FLEXIBILITY OF OPERATION. ITS FUNCTIONS MAY BE INITIATED AUTOMATICALLY AT A POWER UP, OR BY DEPRESSING THE CONSOLE BOOT SWITCH, OR BY A LOAD ADDRESS AND START SEQUENCE.

2. INTERNAL SWITCH SETTING

A SET OF EIGHT MICRO SWITCHES ARE LOCATED ON THE M9312 MODULE. THESE ARE USED BY THE ROUTINES TO DETERMINE WHAT ACTION IS TO BE TAKEN. THEY GIVE THE USER AUTOMATIC ACCESS TO ANY FUNCTION.

A. POWER UP AND CONSOLE BOOT SWITCHES

THE PRIMARY ACTIVATING PROCESSES FOR THE M9312 PROGRAMS ARE EITHER A POWER UP SEQUENCE OR THE ENABLING OF THE CONSOLE BOOT SWITCH.

TO ACTIVATE THE M9312 ON A POWER UP, SWITCH 2 IN THE M9312 MICRO SWITCH REGISTER MUST BE IN THE ON POSITION. IF THIS SWITCH IS OFF THEN A NORMAL TRAP TO LOCATION 24 TO EXECUTE THE USER POWER UP ROUTINE WILL OCCUR. WHEN THIS SWITCH IS ON THE OTHER SWITCHES, 3 THROUGH 10, WILL DETERMINE WHAT ACTION THE M9312 WILL TAKE WHEN THE POWER UP OCCURS (SEE MICRO SWITCH SETTING BELOW).

IF THE SYSTEM INCLUDES A CONSOLE BOOT SWITCH THEN ANY TIME THAT SWITCH IS PRESSED THE M9312 WILL BE ACTIVATED (SOME PROCESSORS MAY HAVE TO BE HALTED FOR THIS SWITCH TO HAVE ANY EFFECT). THE PROCESS USED TO ENTER THE ROM IS A "FAKE" POWER DOWN FOLLOWED BY A POWER UP CAUSED BY PRESSING THE BOOT SWITCH (NOTE THAT THE POSITION OF MICRO SWITCH 2 DESCRIBED ABOVE IS IRRELIVANT TO THE OPERATION OF THIS BOOT SWITCH). THIS RESULTS IN A NORMAL POWER UP SEQUENCE IN THE CPU. PRIOR TO THE POWER UP SEQUENCE, THE M9312 ASSERTS 773000 ON THE UNIBUS ADDRESS LINES. IN THE CASE OF THE 11/60, THE PROCESSOR ASSERTS 000224 ON THE BUS, WHICH IS ORED WITH THE 773000 ASSERTED BY THE M9312. THE 11/70 ASSERTS THE ENTIRE ADDRESS ON THE UNIBUS AS A FUNCTION OF JUMPER SETTINGS (SEE OR M9312 USER MANUAL.) THE NEW PC IS DETERMINED BY ROM LOCATION 773024 OR 773224 (WITH THE 11/60) INSTEAD OF FROM LOCATION 000024. THE NEW PC WILL BE THE LOGICAL 'OR' OF THE CONTENTS OF ROM LOCATION 773024 AND THE EIGHT MICRO SWITCHES ON THE M9312 MODULE (A SWITCH IN THE ON POSITION IS SEEN AS A ONE; LIKEWISE A SWITCH IN THE OFF POSITION IS A ZERO).

IN THIS WAY ALL THE M9312 OPTIONS ARE ACCESSABLE BY MERELY GIVING EACH OPTION A DIFFERENT STARTING ADDRESS. NOTE HERE THAT MICRO SWITCH NUMBER 10 IS OR'ED WITH BIT 1 OF THE DATA IN ROM LOCATION 773024, MICRO SWITCH NUMBER NINE IS OR'ED WITH BIT 2 ETC., AND THAT IT IS UNNECESSARY TO PROVIDE A SWITCH WHICH IS OR'ED WITH DATA BIT 0 AS THIS COULD RESULT IN AN ODD ADDRESS WHEN GOING THROUGH THE TRAP TO LOCATION 773024 SEQUENCE.

B.OPTIONS AND THE MICRO SWITCH SETTINGS.

THE SETTING OF THE MICRO SWITCHS DEPENDS ON THE VARIOUS ROMS ON THE M9312 AND THE DEVICE BOOTS POSTION ON THE M9312.

3. BOOTSTRAPPING

THESE ROUTINES TO BOOTSTRAP A DEVICE TYPICALLY READ IN THE FIRST SECTOR, BLOCK OR 512 (DEC) WORDS, OFF THE DEVICE INTO LOCATION 0 THROUGH 512 (DEC) OF MEMORY. THE EXCEPTIONS TO THIS RULE ARE THE PAPER TAPE BOOT, THE FLEXIBLE DISK BOOT AND THE MAGNETIC TAPE BOOTS. THE PAPER TAPE BOOT IS UNIQUE IN THAT IT CAN DO NO ERROR CHECKING AND THAT THE SECONDARY BOOTSTRAP (THE ABSOLUTE LOADER, FOR EXAMPLE) IS READ INTO THE UPPER PART OF MEMORY. THE ACTUAL LOCATIONS LOADED BY THE PAPER TAPE BOOT ARE PARTIALLY DETERMINED BY THE SECONDARY BOOTSTRAP ITSELF AND BY THE 'SIZE' ROUTINE WHICH DETERMINES THE HIGHEST AVAILABLE MEMORY ADDRESS WITHIN THE FIRST 28K. THE FLEXIBLE DISK (OR FLOPPY) READS SECTOR 1 ON TRACK 1 INTO LOCATIONS STARTING AT ZERO. THE MAGNETIC TAPE BOOTS READ THE SECOND BLOCK INTO LOCATIONS STARTING AT 0. IF NO ERRORS ARE DETECTED IN THE DEVICE, THE BOOTSTRAPS NORMALLY TRANSFER CONTROL TO LOCATION 0 IN ORDER TO EXECUTE THE SECONDARY BOOTSTRAP JUST LOADED. THE ONLY EXCEPTION TO THIS STARTING ADDRESS IS WITH THE PAPER TAPE BOOTS. THEY TRANSFER CONTROL TO LOCATION XXX374, WHERE XXX WAS DETERMINED INITIALLY BY THE SIZE ROUTINE TO BE AT THE TOP OF MEMORY; THIS IS WHERE THE ABSOLUTE LOADER WAS JUST LOADED.

IF A DEVICE ERROR IS DETECTED A RESET WILL BE EXECUTED AND THE BOOTSTRAP WILL TRY AGAIN. THE BOOTSTRAP WILL BE RETRIED INDEFINITELY UNTIL IT SUCCEEDS WITHOUT ERROR UNLESS THE USER (OPERATOR) INTERVENES. THE ADVANTAGE OF RETRYING THE BOOT IS THAT IF A PARTICULAR DEVICE BEING BOOTED IS NOT ONLINE OR LOADED, SAY BECAUSE OF A POWER FAILURE RESTART, THE BOOT WILL GIVE THE DEVICE A CHANCE TO POWER UP (FOR DISKS THIS IS ESSENTIAL). A MAGNETIC TAPE TRANSPORT WILL NOT AUTOMATICALLY RELOAD ITSELF AFTER A POWER FAILURE AND RESTART. THIS SITUATION REQUIRES USER INTERVENTION. THE USER MUST RELOAD THE MAGTAPE AND BRING IT BACK ONLINE AT WHICH TIME THE MAGTAPE BOOTSTRAP, WHICH WAS CONTINUALLY ATTEMPTING TO BOOT THE TAPE, WILL SUCCEED.

NOTE:: AN EXCEPTION TO THIS RULE ARE DEVICE BOOTS FOR THE
RK06/RK07 AND RX02.
IN THE CASE OF THE RK06,IF A DRIVE TYPE ERROR IS
ENCOUNTERED,THE DRIVE IS SET TO AN RK07.
IN THE CASE OF AN RX02,FIRST DOUBLE DENSITY IS TRYED,
SOULD WE GET A DENSITY ERROR ON READ TRY,WE CHANGE
THE DENSITY TO SINGLE DENSITY. IN EITHER CASE,WE
READ TWO (2) SECTORS OF DATA.

SOME BOOTS ALLOW POWER-UP OR BOOT SWITCH BOOTING FOR
DRIVES OTHER THAN DRIVE 0. REFER TO INDIVIDUAL BOOTS FOR THIS
INFORMATION.

4. DIAGNOSTIC TESTS

THERE ARE THREE DIFFERENT TYPES OF TESTS INCLUDED IN
THE M9312 :

1 PRIMARY CPU TESTS
2 SECONDARY CPU TESTS
3 MEMORY TEST
4 CACHE TESTS

THE PRIMARY CPU TESTS ARE TESTS OF MOST UNARY AND
DOUBLE OPERAND INSTRUCTIONS WITH MOST SOURCE MODES. THESE
TESTS DO NOT MODIFY MEMORY. IF A FAILURE IS DETECTED A HALT
WILL OCCUR.

THE SECONDARY CPU TESTS MODIFY MEMORY AND INVOLVE
THE USE OF THE STACK POINTER. THESE TESTS INCLUDE TESTING
OF THE JMP AND JSR INSTRUCTIONS AS WELL AS TESTS OF ALL
DESTINATION MODES. IF A FAILURE IS DETECTED THESE TESTS
WILL ALSO EXECUTE A HALT. THE USER MAY THEN CONSULT THIS
LISTING TO DETERMINE THE FAULT CLASS FOR THE PARTICULAR LOCATION
THE TEST HALTED.

THE MEMORY TEST PERFORMS BOTH A DUAL
ADDRESSING AND DATA CHECK OF ALL THE AVAILABLE MEMORY ON THE
SYSTEM LESS THAN 28K. THIS TEST WILL CHANGE ALL OF MEMORY
TESTED. LIKE THE SECONDARY TESTS THE MEMORY TEST WILL HALT
WHEN AN ERROR IS DETECTED. AT THE TIME THE MEMORY ERROR
HALT IS EXECUTED R0 WILL CONTAIN THE ADDRESS AT WHICH THE
FAILURE WAS DETECTED . R1 WILL CONTAIN THE FAILING DATA
PATTERN AND R0 WILL CONTAIN THE EXPECTED DATA PATTERN.

FINALLY THERE ARE TWO CACHE TESTS WHICH DETERMINE
CPU TYPE (11/60 OR 11/70) AND TEST CACHE CONTROLS, DATA
INTEGRITY AND MEMORY VIA CACHE. FOR DETAILS, SEE TESTS
16 AND 17 OF THE PROGRAM LISTING.

NOTE: HERE THAT DIAGNOSTICS ARE RUN OR NOT RUN (OPTIONALLY) DEPENDING ON WHETHER BIT 1 IS CLEAR OR SET (RESPECTIVELY) IN EITHER:

1 THE INTERNAL MICRO SWITCHES DESCRIBED ABOVE (IF THE POWER UP OR CONSOLE BOOT SWITCH METHOD IS USED); OR

2 THE CONSOLE SWITCH REGISTER (IF THE LOAD ADDRESS AND START WITH OPTION CODE IN SWITCHES METHOD DESCRIBED BELOW IS USED).

5. RESTARTING AT THE USER POWER FAIL ROUTINE

IF THE USER WISHES TO RESTART HIS OWN SOFTWARE ON A POWER UP HE MAY DO SO BY MERELY DISABLING THE POWER FAIL RESTART SWITCH IN THE MICRO SWITCHES (TURN SWITCH 2 OFF).

6. LOAD AND START PROCEDURE

11/60/70 DIAGNOSTIC ROM

THERE ARE NO SPECIAL M9312 SWITCH SETTINGS THAT PERTAIN TO THIS ROM.
THE ONLY WAY THESE DIAGNOSTICS CAN BE EXECUTED IS BY ENTERING A BOOTSTRAP
AT AN ENTRY POINT THAT CALLS FOR DIAGNOSTICS TO BE RUN.

THIS ROM ALLOWS BOOTING VIA CONSOLE SWITCH REGISTER.
THIS CAN BE DONE AS FOLLOWS:

1.LOAD ADDRESS 765744
2.SET SWITCH REGISTER AS SHOWN BELOW

```
! 15 ! 14 ! 13 ! 12 ! 11 ! 10 ! 09 ! 08 ! 07 ! 06 ! 05 ! 04 ! 03 ! 02 ! 01 ! 00 !
!----!----!----!----!----!----!----!----!----!----!----!----!----!----!----!----!
! NA ! NA ! NA ! NA !    !    !    !    !    !    !    !    !    !    !    !    !
!----!----!----!----!----!----!----!----!----!----!----!----!----!----!----!----!
                    ! OCTAL UNIT   !  SWR CODE FROM BOOT ROM TABLES              !
                    !  NUMBER      !                                             !
                    !--------------!---------------------------------------------!
```

3. NOW START

NOTE: THE SWR CODES ARE CONTAINED IN THE BOOT ROM TABLES OF K-SP-M9312-0-4 (SET-UP AND INSTALLATION PROCEDURE.

```
                                          .ABS
        000014                            MISS=14
        000030                            GRP0=30
        000044                            GRP1=44
        165000                            .=165000
165000  010037  000700           START:   MOV     R0,@#700
165004  010137  000702                    MOV     R1,@#702
165010  010437  000704                    MOV     R4,@#704
165014  005037  000706                    CLR     @#706           ;SET FOR 11/70 IF ZERO.
165020  052737  100000  177776            BIS     #100000,@#177776        ;SET BIT 15 IF 11/60,BIT 14 WILL SET
165026  032737  040000  177776            BIT     #40000,@#177776 ;SEE IF 11/60
165034  001402                            BEQ     1$              ;IF NOT SET,11/70
165036  005237  000706                    INC     @#706           ;SET ,WAS AN 11/60
165042  005037  177776           1$:      CLR     @#177776        ;CLR PSW WORD.
                                 ;;****************************************************************
TEST1   THIS TEST VERIFIES THE UNCONDITIONAL BRANCH
                                 ;*
                                 ;*       THE REGISTERS AND CONDITION CODES ARE ALL UNDEFINED WHEN
                                 ;*       THIS TEST IS ENTERED AND THEY SHOULD REMAIN THAT WAY UPON
                                 ;*       THE COMPLETION OF THIS TEST.
                                 ;;****************************************************************
165046                           TST1:
165046  000401                            BR      TST2            ; * BRANCH ALWAYS
165050  000000                            HALT
                                 ;;****************************************************************
TEST2   TEST "CLR", MODE "0", AND "BMI","BVS","BHI","BLT","BLOS
                                 ;*
                                 ;*       THE REGISTERS AND CONDITION CODES ARE ALL UNDEFINED WHEN
                                 ;*       THS TEST IS ENTERED. UPON COMPLETION OF THIS TEST THE "SP"
                                 ;*       (R6) SHOULD BE ZERO AND ONLY THE "Z" FLIP-FLOP WILL BE SET.
                                 ;*
                                 ;;****************************************************************
165052                           TST2:
165052  005006                            CLR     SP              ;N=0,Z=1,V=0,C=0,SP=000000
165054  100404                            BMI     1$              ; V BRANCH IF N=1
165056  102403                            BVS     1$              ; V BRANCH IF V=1
165060  101002                            BHI     1$              ; V BRANCH IF Z AND C ARE BOTH 0
165062  002401                            BLT     1$              ; V BRANCH IF (N XOR V)=1
165064  101401                            BLOS    TST3            ; * BRANCH IF (Z XOR C)=0
165066  000000                   1$:      HALT
                                 ;;****************************************************************
TEST3   TEST "DEC", MODE "0", AND "BPL","BEQ","BGE","BLE"
                                 ;*
                                 ;*       UPON ENTERING THIS TEST THE CONDITION CODES ARE:
                                 ;*       N = 0, Z = 1, V = 0, AND C = 0.
                                 ;*       THE REGISTERS ARE: R0 = ?, R1 = ?, R2 = ?
                                 ;*       R3 = ?, R4 = ?, R5 = ?, SP = 000000
                                 ;*       UPON COMPLETION OF THIS TEST THE CONDITION CODES WILL BE:
                                 ;*       N = 1, Z = 0, V = 0, AND C = 0
                                 ;*       THE REGISTERS AFFECTED BY THE TEST ARE:
                                 ;*       SP = 177777
                                 ;*
                                 ;;****************************************************************
```

```
165070                                TST3:
165070  005306                                DEC     SP              ;N=1,Z=0,V=0,C=0,SP=177777
165072  100003                                BPL     1$              ; V BRANCH IF N=0
165074  001402                                BEQ     1$              ; V BRANCH IF Z=1
165076  002001                                BGE     1$              ; V BRANCH IF (N XOR V)=0
165100  003401                                BLE     TST4            ; * BRANCH IF (Z OR (N XOR V))=1
165102  000000                        1$:     HALT
                                      ;;****************************************************************
TEST4   TEST "ROR, MODE "0", AND "BVC","BHIS","BNE"
                                      ;*
                                      ;*      UPON ENTERING THIS TEST THE CONDITION CODES ARE:
                                      ;*      N = 1, Z = 0, V = 0, AND C = 0.
                                      ;*      THE REGISTERS ARE: R0=?, R1=?, R2 = ?
                                      ;*      R3 = ?, R4 = ?, R5 = ?, SP = 177777
                                      ;*      UPON COMPLETION OF THIS TEST THE CONDITION CODES WILL BE:
                                      ;*      N = 0, Z = 0, V = 1, AND C = 1
                                      ;*      THE REGISTERS AFFECTED BY THE TEST ARE:
                                      ;*      SP = 077777
                                      ;*
                                      ;;****************************************************************
165104                                TST4:
165104  006006                                ROR     SP              ;N=0,Z=0,V=1,C=1,SP=077777
165106  102002                                BVC     1$              ; V BRANCH IF V=0
165110  103001                                BHIS    1$              ; V BRANCH IF C=0
165112  001001                                BNE     TST5            ; * BRANCH IF Z=0
165114  000000                        1$:     HALT
                                      ;;****************************************************************
TEST5   TEST REGISTER DATA PATH
                                      ;*
                                      ;*      WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
                                      ;*      N = 0, Z = 0, V = 1, AND C = 1.
                                      ;*      THE REGISTERS ARE: R0 = ?, R1 = ?, R2 = ?
                                      ;*      R3 = ?, R4 = ?, R5 = ?, SP = 077777.
                                      ;:      UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
                                      ;*      N = 0, Z = 1, V = 0, AND C = 0.
                                      ;*      THE REGISTERS ARE LEFT AS FOLLOWS:
                                      ;*      R0 = 125252, R1 = 000000, R2 = 125252, R3 = 125252
                                      ;*      R4 = 125252, R5 = 125252, AND SP = 125252
                                      ;*
                                      ;;****************************************************************
165116                                TST5:
165116  012706  125252                        MOV     #125252,SP      ;N=0,Z=0,V=0,C=1,SP=125252
165122  010600                                MOV     SP,R0           ;N=0,Z=0,V=0,C=1,R0=125252
165124  010001                                MOV     R0,R1           ;N=0,Z=0,V=0,C=1,R1=125252
165126  010102                                MOV     R1,R2           ;N=0,Z=0,V=0,C=1,R2=125252
165130  010203                                MOV     R2,R3           ;N=0,Z=0,V=0,C=1,R3=125252
165132  010304                                MOV     R3,R4           ;N=0,Z=0,V=0,C=1,R4=125252
165134  010405                                MOV     R4,R5           ;N=0,Z=0,V=0,C=1,R5=125252
165136  160501                                SUB     R5,R1           ;N=0,Z=1,V=0,C=0, AND R1=000000
165140  002401                                BLT     1$              ; V BRANCH IF (N XOR V)=1
165142  001401                                BEQ     TST6            ; * BRANCH IF Z=1
165144  000000                        1$:     HALT
```

```
                                       ;;*****************************************************************
TEST6   TEST "ROL", "BCC", "BLT"
                                       ;*
                                       ;*      WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
                                       ;*      N = 0, Z = 1, V = 0, AND C = 0.
                                       ;*      THE REGISTERS ARE: R0 = 125252, R1 = 000000, R2 = 125252
                                       ;*      R3 = 125252, R4 = 125252, R45 = 125252, SP = 125252.
                                       ;*      UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
                                       ;*      N = 0, Z = 0, V = 1, AND C = 1.
                                       ;*      THE REGISTERS ARE LEFT UNCHANGED EXCEPT FOR
                                       ;*      R2 WHICH SHOULD NOW EQUAL 052524.
                                       ;*
                                       ;;*****************************************************************
165146                                 TST6:
165146  006102                                 ROL     R2              ;N=0,Z=0,V=1,C=1, AND R2 = 052524
165150  103001                                 BCC     1$              ; V BRANCH IF C=0
165152  002401                                 BLT     TST7            ; * BRANCH IF (N XOR V)=1
165154  000000                         1$:     HALT
                                       ;;*****************************************************************
TEST7   TEST "ADD", "INC", "COM", AND "BCS", "BLE"
                                       ;*
                                       ;*      WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
                                       ;*      N = 0, Z = 0, V = 1, AND C = 1.
                                       ;*      THE REGISTERS ARE: R0 = 125252, R1 = 000000, R2 = 052524
                                       ;*      R3 = 125252, R4 = 125252, R5 = 125252, SP = 125252
                                       ;*      UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
                                       ;*      N = 0, Z = 1, V = 0, AND C = 0.
                                       ;*      THE REGISTERS ARE LEFT UNCHANGED EXCEPT FOR
                                       ;*      R3 WHICH NOW EQUALS 000000, AND R1 WHICH IS ALSO 000000
                                       ;*
                                       ;;*****************************************************************
165156                                 TST7:
                                                                       ;(R2 = 052524) + (R3 = 125252)
165156  060203                                 ADD     R2,R3           ;N=1,Z=0,V=0,C=0,AND R3=177776
165160  005203                                 INC     R3              ;N=1,Z=0,V=0,C=0, AND R3=177777
165162  005103                                 COM     R3              ;N=0,Z=1,V=0,C=1, AND R3=000000
165164  060301                                 ADD     R3,R1           ;N=0,Z=1,V=0,C=0, AND R1 = 000000
165166  103401                                 BCS     1$              ; V BRANCH IF C=1
165170  003401                                 BLE     TST10           ; * BRANCH IF (Z OR (N XOR V))=1
165172  000000                         1$:     HALT
                                       ;;*****************************************************************
TEST10  TEST "ROR", "DEC", "BIS", "ADD", AND "BLO"
                                       ;*
                                       ;*      WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
                                       ;*      N = 0, Z = 1, V = 0, AND C = 0.
                                       ;*      THE REGISTERS ARE: R0 = 125252, R1 = 000000, R2 = 052524
                                       ;*      R3 = 000000, R4 = 125252, R5 = 125252, SP = 125252.
                                       ;*      UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
                                       ;*      N = 1, Z = 0, V = 0, AND C = 0.
                                       ;*      THE REGISTERS ARE LEFT UNCHANGED EXCEPT FOR
                                       ;*      R4 WHICH SHOULD NOW EQUAL 052525, AND
                                       ;*      R1 WHICH SHOULD NOW EQUAL 177777
                                       ;*
                                       ;;*****************************************************************
```

```
165174                          TST10:
165174  006004                          ROR     R4              ;N=0,Z=0,V=1,C=0, AND R4 = 052525
165176  050403                          BIS     R4,R3           ;N=0,Z=0,V=0,C=0, AND R3 = 052525
165200  060503                          ADD     R5,R3           ;N=1,Z=0,V=0,C=0, AND R3 = 177777
165202  005203                          INC     R3              ;N=0,Z=1,V=0,C=0, AND R3 = 000000
165204  103402                          BLO     1$              ; V BRANCH IF C=1
165206  005301                          DEC     R1              ;N=1,Z=0,V=0,C=0, AND R1 = 177777
165210  002401                          BLT     TST11           ; * BRANCH IF (N XOR V)=1
165212  000000                  1$:     HALT
                                ;*
                                ;;*****************************************************************
TEST11  TEST "COM", "BIC", AND "BGT", "BLE"
                                ;*
                                ;*      WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
                                ;*      N = 1, Z = 0, V = 0, AND C = 0.
                                ;*      THE REGISTERS ARE: R0 = 125252, R1 = 177777, R2 = 052524
                                ;*      R3 = 000000, R4 = 052525, R5 = 125252, SP = 125252.
                                ;*      UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
                                ;*      N = 0, Z = 0, V = 1, AND C = 1.
                                ;*      THE REGISTERS ARE LEFT UNCHANGED EXCEPT FOR
                                ;*      R0 WHICH SHOULD NOW EQUAL 052525, AND
                                ;*      R1 WHICH SHOULD NOW EQUAL 052524
                                ;*
                                ;;*****************************************************************
165214                          TST11:
165214  005100                          COM     R0              ;N=0,Z=0,V=0,C=1, AND R0 = 052525
165216  101401                          BLOS    2$              ; * BRANCH IF (Z OR C)=1
165220  000000                          HALT                    ;STOP HERE IF BRANCH FAILED
165222  040001                  2$:     BIC     R0,R1           ;N=1,Z=0,V=0,C=1, AND R1 = 125252
165224  060101                          ADD     R1,R1           ;N=0,Z=0,V=1,C=1, AND R1 = 052524
165226  003001                          BGT     1$              ; V BRANCH IF Z AND ( N XOR V) ARE BOTH 0
165230  003401                          BLE     TST12           ; * BRANCH IF (Z OR (N XOR V))=1
165232  000000                  1$:     HALT
                                ;;*****************************************************************
TEST12  TEST "SWAB", "CMP", "BIT", AND "BNE", "BGT"
                                ;*
                                ;*      WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
                                ;*      N = 0, Z = 0, V = 1, AND C = 1.
                                ;*      THE REGISTERS ARE: R0 = 052525, R1 = 052524, R2 = 052524
                                ;*      R3 = 000000, R4 = 052525, R5 = 125252, SP = 125252.
                                ;*      UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
                                ;*      N = 0, Z = 0, V = 0, AND C = 1.
                                ;*      THE REGISTERS ARE NOW:
                                ;*      R0 = 052525, R1 = 052125, R2 = 052524, R3 = 000000
                                ;*      R4 = 052525, R5 = 052525, SP = 125252.
                                ;*
                                ;;*****************************************************************
165234                          TST12:
165234  000301                          SWAB    R1              ;N=0,Z=0,V=0,C=0, AND R1 = 052125
165236  020127  052125                  CMP     R1,#052125              ;N=0,Z=1,V=0,C=0
165242  001004                          BNE     1$              ; V BRANCH IF Z=0
                                                                ;R4 = 052525 R5 = 125252
165244  030405                          BIT     R4,R5           ;N=0,Z=1,V=0,C=0
165246  003002                          BGT     1$              ; V BRANCH IF Z OR (N XOR V) ARE 0
165250  005105                          COM     R5              ;N=0,Z=0,V=0,C=1, AND R5 = 052525
```

```
165252  001001                          BNE     TST13           ; * BRANCH IF Z=1
165254  000000                  1$:     HALT
                                ;;*******************************************************************
TEST13  TEST "MOVB", "SOB", "CLR", "TST" AND "BPL", "BNE"
                                ;*
                                ;*      WHEN THIS TEST IS ENTERED THE CONDITION CODES ARE:
                                ;*      N = 0, Z = 0, V = 0, AND C = 1.
                                ;*      THE REGISTERS ARE: R0 = 052525, R1 = 052125, R2 = 052524
                                ;*      R3 = 000000, R4 = 052525, R5 = 052525, SP = 125252.
                                ;*      UPON COMPLETION OF THIS TEST THE CONDITION CODES ARE:
                                ;*      N = 0, Z = 1, V = 0, AND C = 0.
                                ;*      R0 IS DECREMENTED BY A SOB INSTRUCTION TO 000000
                                ;*      R1 IS CLEARED AND THEN INCREMENTED AROUND TO 000000
                                ;*
                                ;;*******************************************************************
165256                          TST13:
165256  112700  177401                  MOVB    #177401,R0      ;N=0,Z=0,V=0,C=1, AND R0 = 000001
165262  100001                          BPL     2$              ; * BRANCH IF N=0
165264  000000                  1$:     HALT                    ;STOP IF "BPL" FAILED
165266  077002                  2$:     SOB     R0,1$           ;DO NOT LOOP SINCE (R0 -1) = 0
165270  005001                          CLR     R1              ;N=0, Z=1, V-0, C=0, AND R1 = 000000
165272  005201                  3$:     INC     R1              ;INCREMENT 64K TIMES (2 ** 16)
165274  077002                          SOB     R0,3$           ;LOOP BACK TO "INC" 64K TIMES
165276  005700                          TST     R0              ;N=0,Z=1,V=0,C=0, AND R0 = 000000
165300  001002                          BNE     4$              ; V BRANCH IF Z=0
165302  005701                          TST     R1              ;N=0,Z=1,V=0,C=0, AND R1 = 000000
165304  001401                          BEQ     TST14
165306  000000                  4$:     HALT
                                ;;*******************************************************************
TEST14  TEST "JSR", "RTS", "RTI", _"JMP"
                                ;*
                                ;*      THIS TEST FIRST SETS THE STACK POINTER TO 776,
                                ;*      AND THEN VERIFIES THAT "JSR", "RTS", "RTI", AND "JMP"
                                ;*      ALL WORK PROPERLY.
                                ;*
                                ;*      ON ENTRY TO THIS TEST THE STACK POINTER "SP" IS INITIALIZED
                                ;*      TO 00776 AND IS LEFT THAT WAY ON EXIT.
                                ;*
                                ;;*******************************************************************
165310                          TST14:
165310  012706  000776          11$:    MOV     #776,SP         ;SET UP THE STACK POINTER
165314  004767  000002                  JSR     PC,1$           ;TRY TO JSR TO 1$
165320  000000                  10$:    HALT                    ;THE "JSR" MUST HAVE FAILED
165322  022716  165320                  1$:     CMP     #10$,(SP)       ;WAS THE CORRECT ADDRESS PUSHED?
165326  001401                          BEQ     2$              ;BRANCH IF YES
165330  000000                          HALT                    ;WRONG THING PUSHED ON STACK
165332  012716  165342          2$:     MOV     #3$,(SP)        ;CHANGE THE ADDRESS ON THE STACK
165336  000207                          RTS     PC              ;TRY TO RETURN TO 3$
165340  000000                          HALT                    ;DID NOT RETURN PROPERLY
165342  005046                  3$:     CLR     -(SP)           ;PUSH A ZERO ON THE STACK
165344  012746  165354                  MOV     #4$,-(SP)       ;PUSH THE RETURN ADDRESS ON STACK
165350  000002                          RTI                     ;SEE IF AN "RTI" WORKS
165352  000000                          HALT                    ;THE "RTI" FAILED
165354  000137  165362          4$:     JMP     @#5$            ;TRY TO "JMP"
165360  000000                          HALT                    ;THE "JMP" FAILED
```

```
165362                          5$:                                     ;ADDRESS TO "JMP" TO
                                ;;*****************************************************************
TEST15  TEST MAIN MEMORY FROM VIRTUAL 001000 TO LAST ADDR.
                                ;*
                                ;*      THIS TEST WILL TEST MAIN MEMORY WITH THE CACHE DISABLED, FROM
                                ;*      VIRTUAL ADDRESS 001000 TO LAST ADDR. IF THE DATA DOES NOT COMPARE
                                ;*      PROPERLY THE TEST WILL HALT AT EITHER 165516 OR 165536. IF A
                                ;*      PARITY ERROR OCCURS THE TEST WILL HALT AT ADDRESS 165750, WITH
                                ;*      THE PC + 2 ON THE STACK WHICH IS IN THE KERNEL D-SPACE.
                                ;*
                                ;*      IN THIS TEST THE REGISTERS ARE INITIALIZED AS FOLLOWS:
                                ;*      R0 = 001000, R1 = DATA READ, R2 = 001000, R3 = 177746 (CACHE CONTROL REG.)
                                ;*      R4 = COUNT VALUE, R5 = LAST MEMORY ADDRESS, SP = 000776
                                ;*
                                ;;*****************************************************************
165362                          TST15:
165362  012705  160000                  MOV     #160000,R5      ;FIRST GET SIZE OF MEMORY.
165366  005037  000006                  CLR     @#6
165372  012737  165400  000004          MOV     #1$,@#4
165400  012706  000776          1$:     MOV     #776,SP
165404  005745                          TST     -(R5)
                                                                ;IN END,R5 CONTAINS LAST MEM ADDR.
165406                          FIX:
                                                                ;THE LAST MEMORY ADDRESS
165406  012737  165714  000114  10$:    MOV     #CONT,@#114     ;SET UP PARITY VECTOR
165414  005037  000116                  CLR     @#116           ;SET PROCESSOR STATUS WORD TO ZERO
165420  012703  177746                  MOV     #177746,R3      ;CACHE CONTROL REGISTER ADDRESS
165424  012713  000014                  MOV     #MISS,(R3)      ;FORCE MISS BOTH GROUPS
165430  012702  001000                  MOV     #1000,R2        ;FIRST ADDRESS STORAGE
165434  010200                          MOV     R2,R0           ;SETUP FORST ADDRESS
165436  010010                  1$:     MOV     R0,(R0)         ;LOAD EACH ADDRESS WITH ITS
                                                                ;OWN ADDRESS
165440  005720                  TST     (R0)+
165442  020005                          CMP     R0,R5
165444  101774                          BLOS    1$
165446  010200                          MOV     R2,R0           ;SET STARTING ADDRESS IN R0
165450  011001                  2$:     MOV     (R0),R1         ;GET THE DATA
165452  020001                          CMP     R0,R1           ;IS IT CORRECT?
165454  001401                          BEQ     3$              ;BRANCH IF YES
165456  000000                          HALT                    ;DATA ERROR ON READING MEMORY LOCATION
                                                                ;R0 = ADDRESS, R1 = DATA RECEIVED, R0 = DATA EXPECTED
165460  005120                  3$:     COM     (R0)+           ;COMPLEMENT DATA AND INCREMENT ADDRESS
165462  020005                          CMP     R0,R5
165464  101771                          BLOS    2$
165466  014001                  4$:     MOV     -(R0),R1        ;READ THE DATA (IT SHOULD NOW BE THE
                                                                ;COMPLEMENT OF THE ADDRESS)
165470  005101                          COM     R1              ;COMPLEMENT BEFORE CHECKING
165472  020001                          CMP     R0,R1           ;IS THE DATA CORRECT?
165474  001401                          BEQ     5$              ;BRANCH IF YES
165476  000000                          HALT                    ;DATA ERROR ON READING MEMORY LOCATION
                                                                ;R0=ADDRESS, R1=DATA RECEIVED, R0=DATA EXPECTED
165500  020002                  5$:     CMP     R0,R2
165502  001371                          BNE     4$
```

```
        CACHE MEMORY DIAGNOSTIC TESTS
                                                   ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
                                                   ;VVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVVV
                                                   ;*
                                                   ;*      THE FOLLOWING TWO TESTS ARE CACHE MEMORY TESTS, IF EITHER OF
                                                   ;*      THEM FAILS TO RUN SUCCESSFULLY THEY WILL COME TO A HALT
                                                   ;*      IN THE M9312 DIAGNOSTIC. IF YOU DESIRE TO TRY TO BOOT YOUR SYSTEM, OR
                                                   ;*      DIAGNOSTIC ANYWAY, YOU CAN PRESS "CONTINUE" AND THE PROGRAM
                                                   ;*      WILL FORCE MISSES IN THE CACHE AND GO TO THE
                                                   ;*      BOOT STRAP THAT HAS BEEN SELECTED.
                                                   ;*
                                                   ;
                                                   ;
                                                   ;;**********************************************************************
        TEST16  TEST CACHE DATA MEMORY
                                                   ;*
                                                   ;*      THIS TEST WILL CHECK THE DATA MEMORY IN THE CACHE, ON THE PDP11/60
                                                   ;       THERE IS ONLY ONE GROUP (1-K), ON PDP11/70 THERE ARE
                                                   ;       TWO GROUPS, 1/2K EACH.
                                                   ;*      THE TEST LOADS 0552525 INTO AN ADDRESS, COMPLEMENTS
                                                   ;*      IT TWICE AND THEN READS THE DATA, THEN IT CHECKS TO INSURE THAT
                                                   ;*      THE DATA WAS A HIT. THEN THE SEQUENCE IS REPEATED ON THE SAME
                                                   ;*      ADDRESS WITH 125252 AS THE DATA. ALL CACHE MEMORY DATA LOCATIONS
                                                   ;*      ARE TESTED IN THIS WAY.
                                                   ;*      IF EITHER GROUP FAILS AND THE OPERATOR PRESSES CONTINUE THE
                                                   ;*      PROGRAM WILL TRY TO BOOT WITH THE CACHE DISABLED.
                                                   ;*
                                                   ;*      THE REGISTERS ARE INITIALIZED AS FOLLOWS FOR THIS TEST:
                                                   ;*      R0 = 4000 (ADDRESS), R1 = 2 (COUNT), R2 = 1000 (COUNT)
                                                   ;*      R3 = 177746 (CONTROL REG.), R4 = 125252 (PATTERN), R5 = LAST MEMORY ADDRESS
                                                   ;*      SP = 000776 (FLAG OF ZERO PUSHED ON STACK)
                                                   ;*
                                                   ;;**********************************************************************
        165504                                     TST16:
        165504  005016                                     CLR     (SP)            ;SET THE CYCLE FLAG TO ZERO, PATTERN FLAG TO 0
        165506  012704  125252                             MOV      125252,R4      ;SET UP R4 FOR TEST
        165512  012713  000030                             MOV      GRP0,(R3)      ;FORCE REPLACE FROUP 0 AND FORCE MISS GROUP 1 (ON 11/70)
                                                                                   ;FOR MISS UPPER 1/2K OF CACHE ON 11/60
        165516  012700  004000                             MOV      4000,R0        ;SET STARTING ADDRESS INTO R0
        165522  012702  001000                     1$:     MOV      1000,R2        ;SET COUNT TO 1000 OCTAL
        165526  005104                             3$:     COM     R4              ;COMPLEMENT DATA IN R4
        165530  010410                                     MOV     R4,(R0)         ;WRITE THE TEST PATTERN
        165532  005110                                     COM     (R0)            ;DOUBLE COMPLEMENT DATA AND
        165534  005110                                     COM     (R0)            ;MAKE SURE DATA IS IN THE CACHE
        165536  021004                                     CMP     (R0),R4         ;COMPARE DATA _SET BIT0 IN HIT/MISS REG.
        165540  001401                                     BEQ     5$              ;BRANCH IF DATA MATCHES
        165542  000000                                     HALT                    ;CACHE DATA DIDN'T MATCH
                                                                                   ;R0=ADDRESS, R4=EXPECTED DATA
        165544  006037  177752                     5$:     ROR     @ 177752        ;WAS THE LAST MEMORY REFERENCE A HIT?
        165550  103402                                     BCS     4$              ;BRANCH IF YES
        165552  000000                                     HALT                    ;CACHE FAILED TO HIT
                                                                                   ;R0=ADDRESS THAT WAS REFERENCED
        165554  000461                                     BR      BOOTMISS        ;ABROT REST OF TEST IF "CONTINUE" PRESSED
        165556  105116                             4$:     COMB    (SP)
        165560  001362                                     BNE     3$
```

```
165562  000402                          BR      6$
165564  000167  177210                  JMP     START           ;ENTRY POINT FROM ROM WITH DIAGNOSTIC SELECTED
165570  005720                  6$:     TST     (R0)+           ;MOVE TO NEXT ADDRESS
165572  077223                          SOB     R2,3$           ;BRANCH IF NOT DONE
165574  012713  000044                  MOV     #GRP1,(R3)      ;FORCE MISS LOWER 1/2K OF CACHE ON 11/60
165600  012700  006000                  MOV     #6000,R0
165604  105166  000001                  COMB    1(SP)           ;COMPLEMENT THE CYCLE FLAG
165610  001344                          BNE     1$              ;LOOP IF NOT DONE
                                ;;*****************************************************************
TEST17  TEST MEMORY WITH THE DATA CACHE ON
                                ;*
                                ;*      THIS TEST CHECKS VIRTUAL MEMORY FROM 001000 THRU LAST ADDRESS
                                ;*      TO INSURE THAT YOU CAN GET HITS ALL THE WAY UP THROUGH MAIN
                                ;*      MEMORY. ON THE PDP11/70, IT STARTS WITH GROUP 1 ENABLED, THEN TESTS
                                ;*      GROUP 0, AND FINALLY CHECKS MEMORY WITH BOTH GROUPS ENABLED.
                                ;*      ON THE PDP11/60, THE TEST IS DONE WITH THE
                                ;*      WHOLE CACHE ENABLED.
                                ;*
                                ;*      UPON ENTRY THE REGISTERS WILL BE SET UP AS FOLLOWS:
                                ;*
                                ;*      R0 = 001000 (ADDRESS), R1 = 3 (PASS COUNT), R2 = (FIRST ADDRESS),
                                ;*      R3 = 177746 (CONTROL REG.),
                                ;*      R5 = (LAST MEMORY ADDRESS, SP = 776
                                ;*
                                ;*      UPON COMPLETION OF THIS TEST MAIN MEMORY FROM VIRTUAL ADDRESS
                                ;*      001000 THRU LAST ADDRESS WILL CONTAIN ITS OWN VIRTUAL ADDRESS.
                                ;;*****************************************************************
165612                          TST17:
165612  012702  001000                  MOV     #1000,R2        ;SETUP FIRST ADDRESS
165616  010200                          MOV     R2,R0           ;FIRST ADDRESS IS 1000 OCTAL
165620  010010                  1$:     MOV     R0,(R0)         ;FILL MEMORY WITH ADDRESSES
165622  005720                          TST     (R0)+
165624  020005                          CMP     R0,R5
165626  101774                          BLOS    1$
165630  012701  000003                  MOV     #3,R1           ;SET PASS COUNT TO THREE
165634  005016                          CLR     (SP)
165636  005737  000706                  TST     @#706           ;IF 11/60 THIS LOC=1,11/70=00
165642  001020                          BNE     6$              ;IT IS PDP11/60
165644  012716  000030                  MOV     #GRP0,(SP)      ;LOAD CODE TO FORCE GROUP 0 ONTO STACK
165650  010200                  2$:     MOV     R2,R0           ;FIRST ADDRESS
165652  005110                  3$:     COM     (R0)            ;DOUBLE COMPLEMENT DATA AND
165654  005110                          COM     (R0)            ;MAKE SURE IT IS IN THE CACHE.
165656                          51$:
165656  020010                          CMP     R0,(R0)         ;COMPARE DATA, AND SET BIT 0 IN HIT/MISS REG.
                                                                ;ALSO POINT TO NEXT ADDRESS
165660  001401                          BEQ     5$              ;BRANCH IF DATA MATCHES
165662  000000                          HALT                    ;DATA DIDN'T MATCH R0 = ADDRESS + 2
165664  005720                  5$:     TST     (R0)+
165666  006037  177752                  ROR     @#177752        ;WAS THE LAST MEMORY REFERENCE A HIT?
165672  103402                          BCS     4$              ;BRANCH IF YES
165674  000000                          HALT                    ;HIT FAILED TO OCCUR R0 = ADDRESS + 2
165676  000410                          BR      BOOTMISS        ;ABORT REST OF TEST IF "CONTINUE" PRESSED
165700  020005                  4$:     CMP     R0,R5
165702  101763                          BLOS    3$
165704  011613                  6$:     MOV     (SP),(R3)       ;FORCE MISS GRP1 ON PASS 2, FULLY
```

```
                                                                    ;ENABLE CACHE ON PASS THREE. (11/70)
                                                                    ;ON 11/60, RUN EACH PASS WITH THE SHOLE
                                                                    ;CACHE ENABLED.
   165706  005016                           CLR     (SP)            ;GET READY TO FULLY ENABLE CACHE ON PASS 3
   165710  077121                           SOB     R1,2$           ;RUN THREE PASSES THRU THIS TEST
   165712  000404                   JUMP0:  BR      JUMP            ;GO TO BOOT STRAP CODE
   165714  000000                   CONT:   HALT                    ;STOP HERE IF THERE IS A CACHE PARITY ERROR
                                                                    ;OR A MAIN MEMORY PARITY ERROR
                                                                    ;CHECK CCR, MEMORY REGISTER AND CPU REGISTER
                                                                    ;TO FIND WHICH ONE
   165716  000402                           BR      JUMP
   165720                           BOOTMISS:
   165720  012713  000014                   MOV     #MISS,(R3)      ;FORCE MISSES IN BOTH GROUPS OF CACHE
   165724  013700  000700         JUMP:    MOV     @#700,R0        ;NOW RESTORE ALL NEEDED REGISTERS.
   165730  013701  000702                   MOV     @#702,R1        ;AND RETURN TO BOOTSTRAP.
   165734  013704  000704                   MOV     @#704,R4
   165740  000164  000002                   JMP     2(R4)
TEST17     TEST MEMORY WITH THE DATA CACHE ON
   165744  013704  177570                   MOV     @#177570,R4     ;SWITCH REGISTER TO R4
                                                                    ;OFFSET ADDR.IN BITS 0-8 OF R4
   165750  042704  177000                   BIC     #177000,R4
   165754  052704  173000                   BIS     #173000,R4      ;BITS 0-8 OF SWR CONTAINS STARTING CODE.
                                                                    ;OF DESIR"" ROM.
   165760  113700  177571                   MOVB    @#177571,R0     ;BITS 9-1  )F SWR = OCTAL UNIT NUMBER.
   165764  006200                           ASR     R0              ;UNIT # RIGHT JUSTIFED IN R0
   165766  000241                           CLC                     ;MAKE SURE C BIT CLEAR
   165770  000114                           JMP     (R4)            ;EXIT TO BOOT ROM
   165772  000000                           HALT                    ;EXTRA WORD
   165774  041060                           .ASCII  "0B"            ;IDENTIFIES ROM AS "B0"   OR B VERSION REV 0.  ECB 12-12
   165776  025055                           .WORD   025055          ;CONTAINS CRC-16 WORD FOR LAST 255 WORDS.
           000001                           .END
```

```
                          616F1.M11    SYMBOL TABLE

BIT8   = 000400      BIT9   = 001000      BOOTMI  165720      CONT    165714
CRCWD  = 000000      DIAG   = 165564      FIX     165406      GRP0  = 000030
GRP1   = 000044      HSRCR  = 177550      INITSW= 173024      JUMP    165724
JUMP0    165712      MISS   = 000014      MRESER= 173000      PC    =%000007
RC11CR= 177446      RESERV= 000340      RF11CR= 177460      RK05CR= 177404
RK06CR= 177440      RL01CR= 174400      RP03CR= 176714      RP04CR= 176700
RS03CR= 172040      RS04CR= 172040      RX01CR= 177170      RX02CR= 177170
R0     =%000000      R1     =%000001      R2     =%000002      R3    =%000003
R4     =%000004      R5     =%000005      R6     =%000006      R7    =%000007
SP     =%000006      START    165000      TST1     165046      TST10   165174
TST11    165214      TST12    165234      TST13    165256      TST14   165310
TST15    165362      TST16    165504      TST17    165612      TST2    165052
TST3     165070      TST4     165104      TST5     165116      TST6    165146
TST7     165156      TTCR   = 177560      TU10CR= 172522      TU16CR= 172440
TU56CR= 177342      .      = 166000
```